universidade federal de pelotas
CDTec
centro de desenvolvimento tecnológico

computação
Universidade Federal de Pelotas

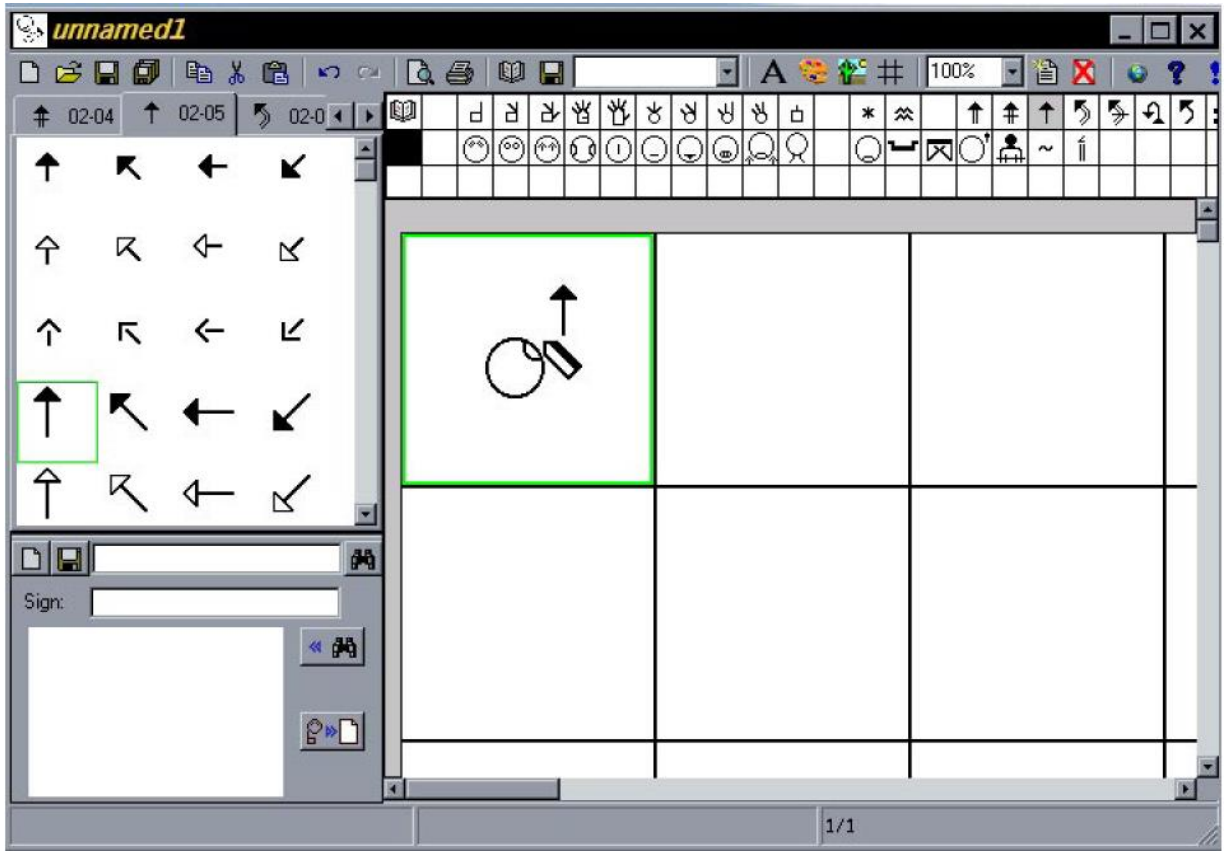
Programação Avançada
Hoje: Introdução

Prof. Dr. Rafael P. Torchelsen
rafael.torchelsen@inf.ufpel.edu.br

Prof. Dr. Rafael P. Torchelsen

computação
Universidade Federal de Pelotas

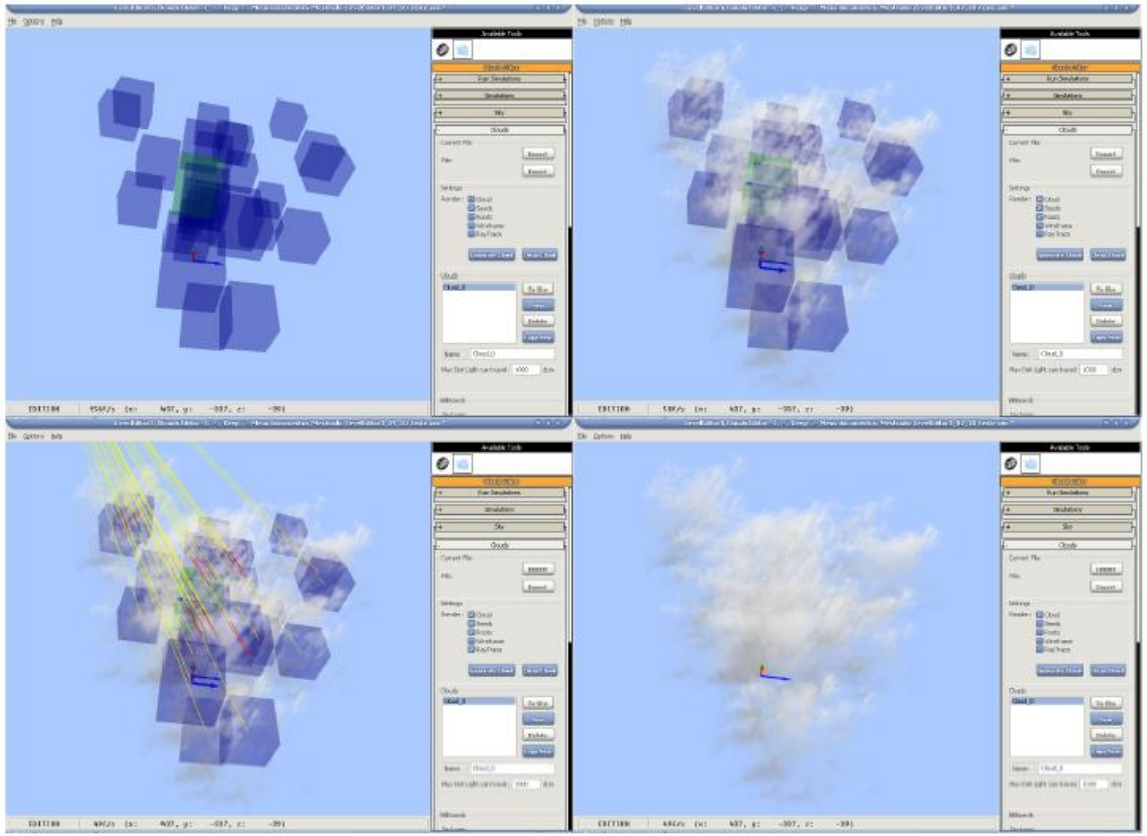
- **Graduação**: Universidade Católica de Pelotas - **UCPEL**



Prof. Dr. Rafael P. Torchelsen

computação
Universidade Federal de Pelotas

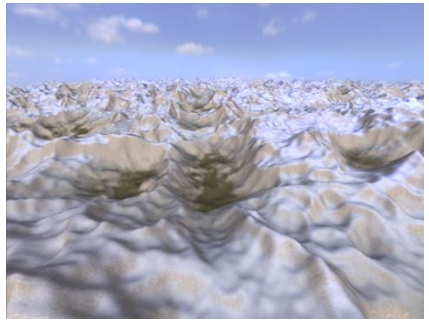
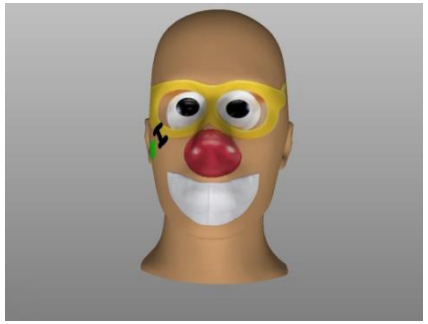
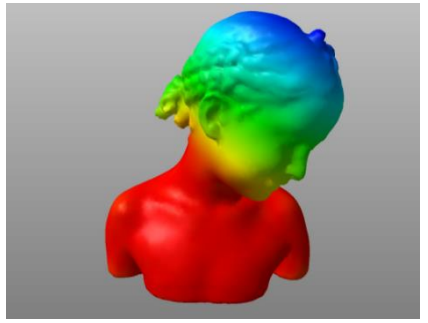
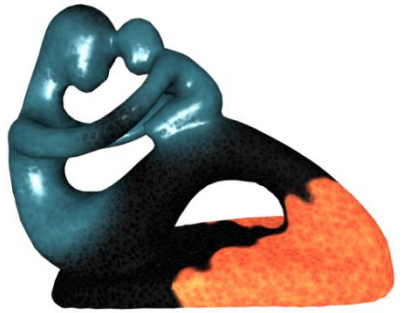
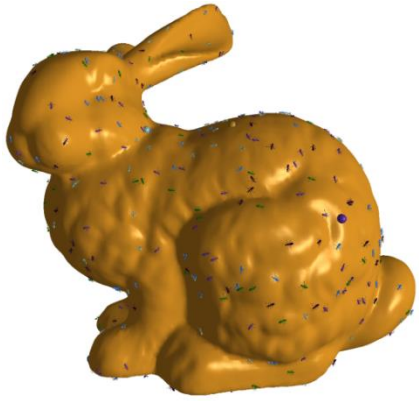
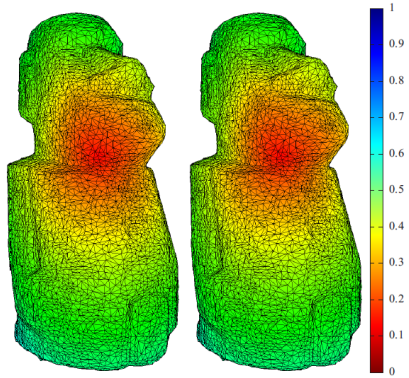
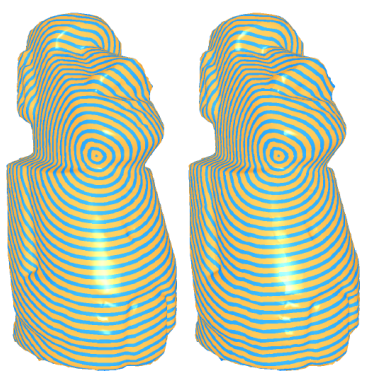
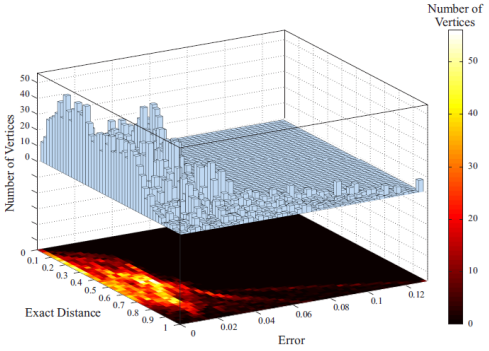
- **Graduação**: Universidade Católica de Pelotas - **UCPEL**
- **Mestrado**: Universidade do Vale do Rio dos Sinos – **UNISINOS**



Prof. Dr. Rafael P. Torchelsen

computação
Universidade Federal de Pelotas

- **Graduação**: Universidade Católica de Pelotas - **UCPEL**
- **Mestrado**: Universidade do Vale do Rio dos Sinos – **UNISINOS**
- **Doutorado**: Universidade Federal do Rio Grande do Sul – **UFRGS**

Prof. Dr. Rafael P. Torchelsen

computação
Universidade Federal de Pelotas

- **Graduação**: Universidade Católica de Pelotas - **UCPEL**
- **Mestrado**: Universidade do Vale do Rio dos Sinos – **UNISINOS**
- **Doutorado**: Universidade Federal do Rio Grande do Sul – **UFRGS**
- **Mercado:** Desenvolvimento de jogos – **SouthLogic**

Prof. Dr. Rafael P. Torchelsen

computação
Universidade Federal de Pelotas

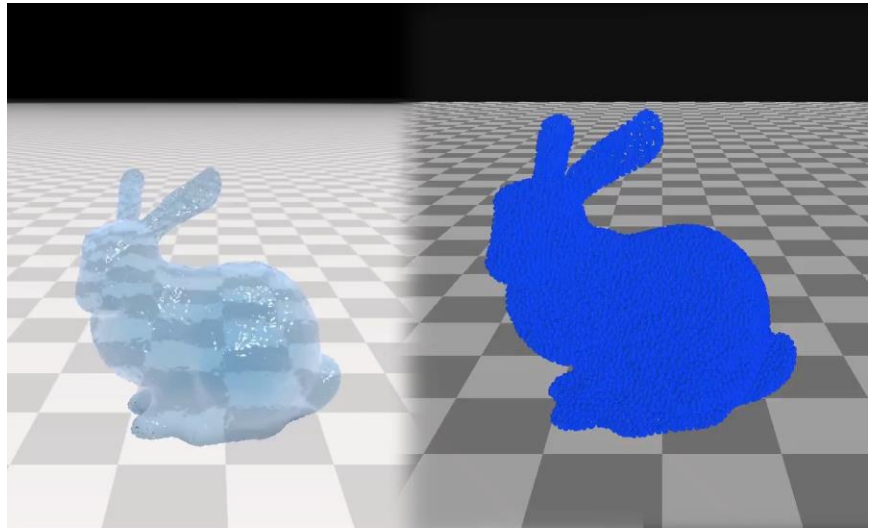
- **Graduação**: Universidade Católica de Pelotas - **UCPEL**
- **Mestrado**: Universidade do Vale do Rio dos Sinos – **UNISINOS**
- **Doutorado**: Universidade Federal do Rio Grande do Sul – **UFRGS**
- **Mercado:** Desenvolvimento de jogos – **SouthLogic**
- **Docente:** Universidade Federal da Fronteira Sul – **UFFS**
- **Docente**: Universidade Federal de Pelotas - **UFPel**

UFFS

UNIVERSIDADE FEDERAL DE PELOTAS
RS - BRASIL

Pesquisa

computação
Universidade Federal de Pelotas



Link

PHYS-SKETCH
Sketching 3D Dynamic Objects in Immersive Virtual Reality

Link

Real-Time Local Unfolding for
Agents Navigation on Arbitrary Surfaces
Sibgrapi paper ID: 46

Link

Efficient Surgical Cutting with Position-based Dynamics

Link

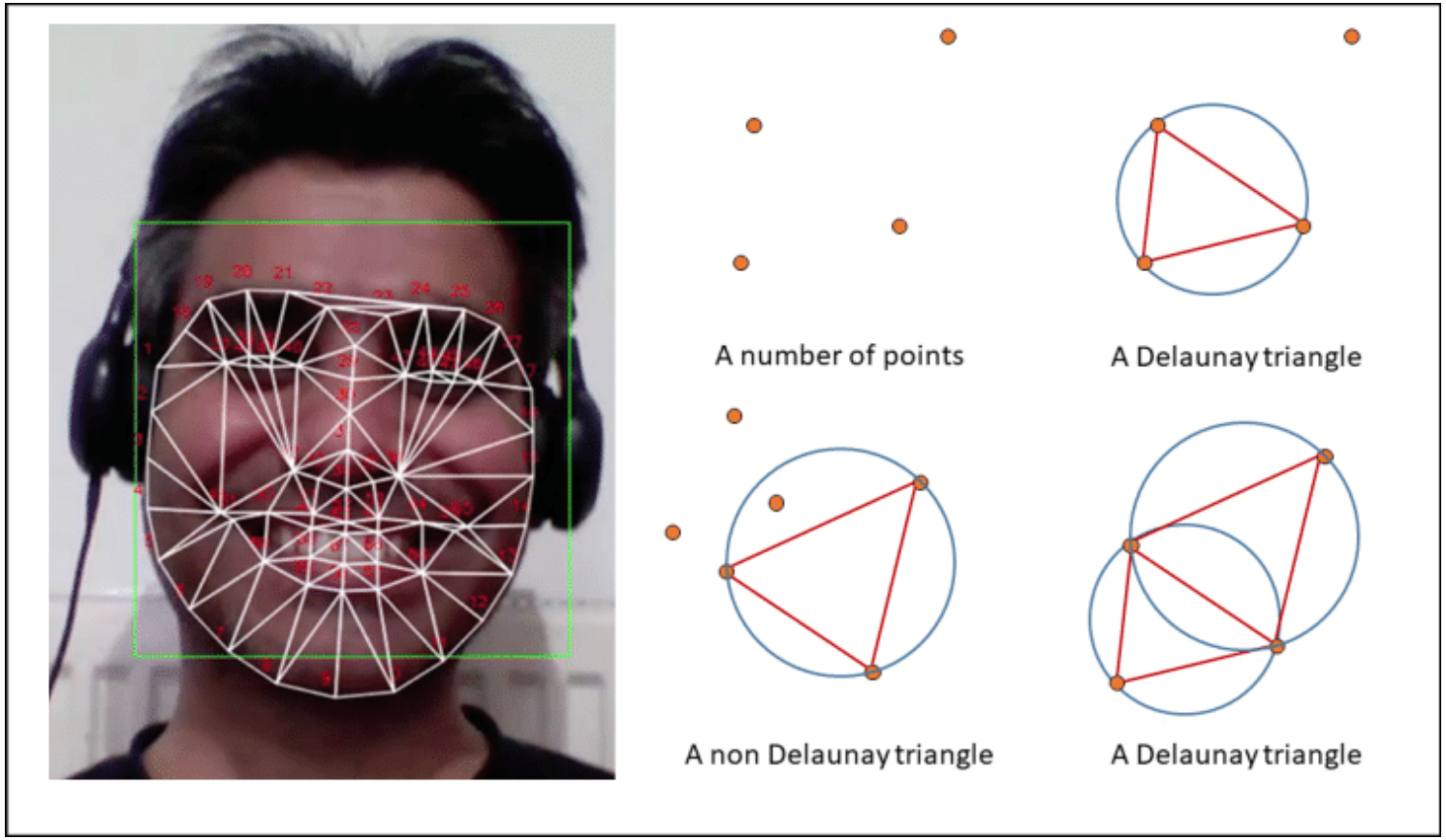
Programação Avançada

computação
Universidade Federal de Pelotas

- O avançada é por vermos tópicos que normalmente não são vistos na graduação em Computação.
- Tópicos:
  - Geometria computacional
  - Padrões de projeto
  - Programação orientada a dados
  - Simulação física
  - etc

Geometria computacional: Reconhecimento facial

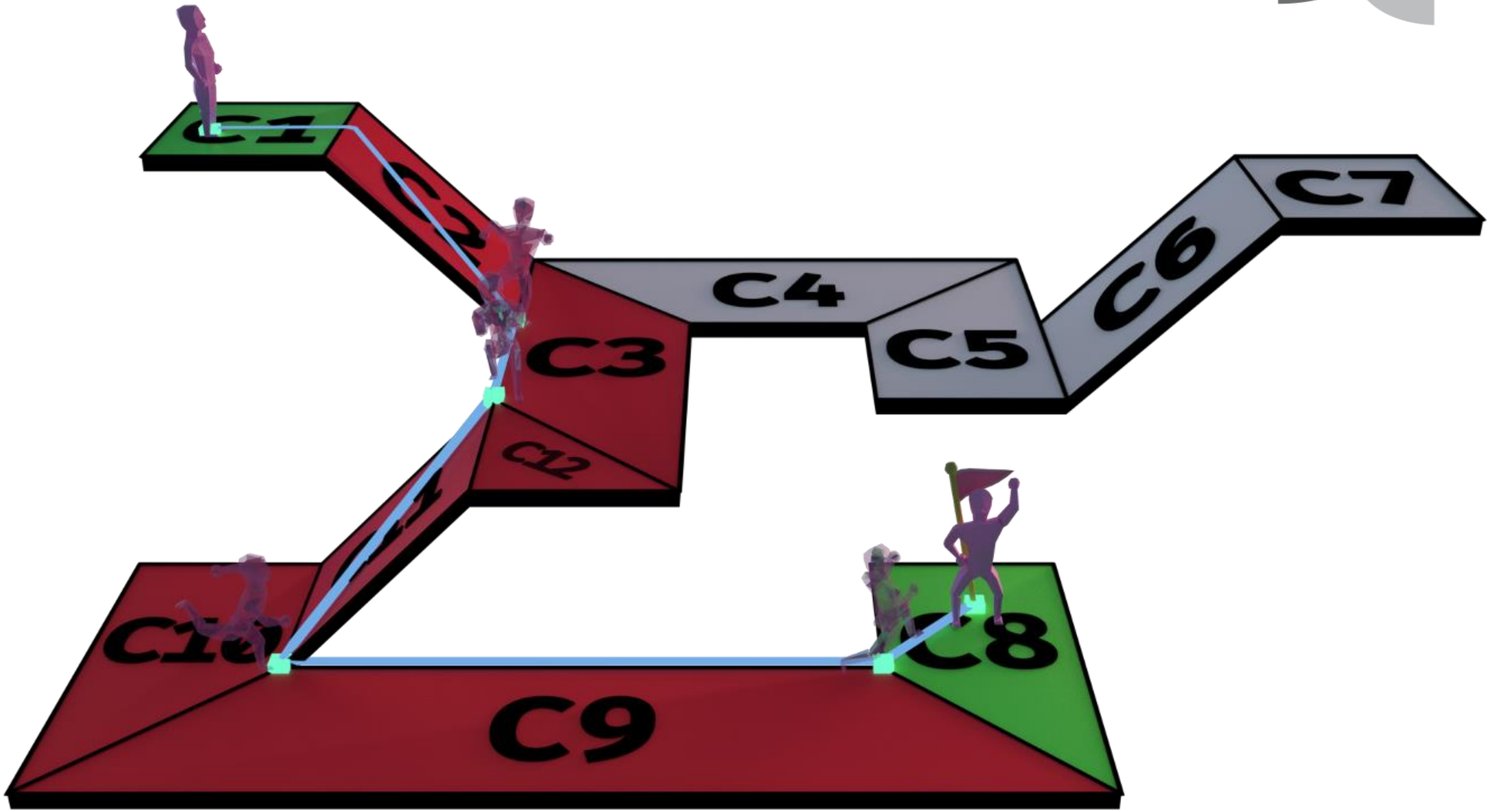
computação
Universidade Federal de Pelotas



Triangulação de Delaunay

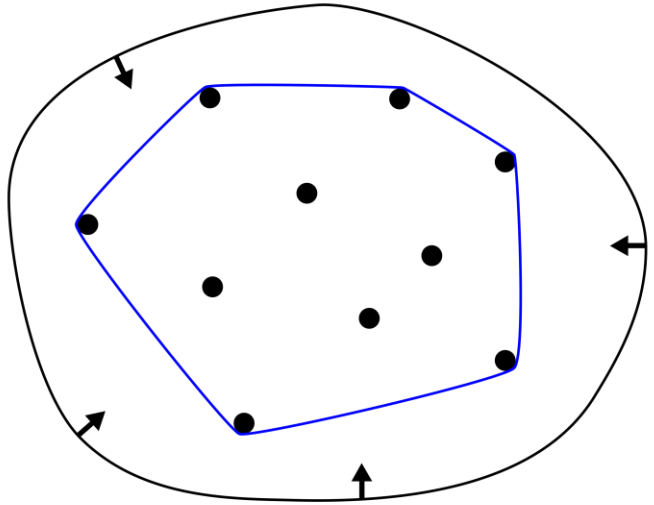
Geometria computacional: Busca de caminho

computação
Universidade Federal de Pelotas

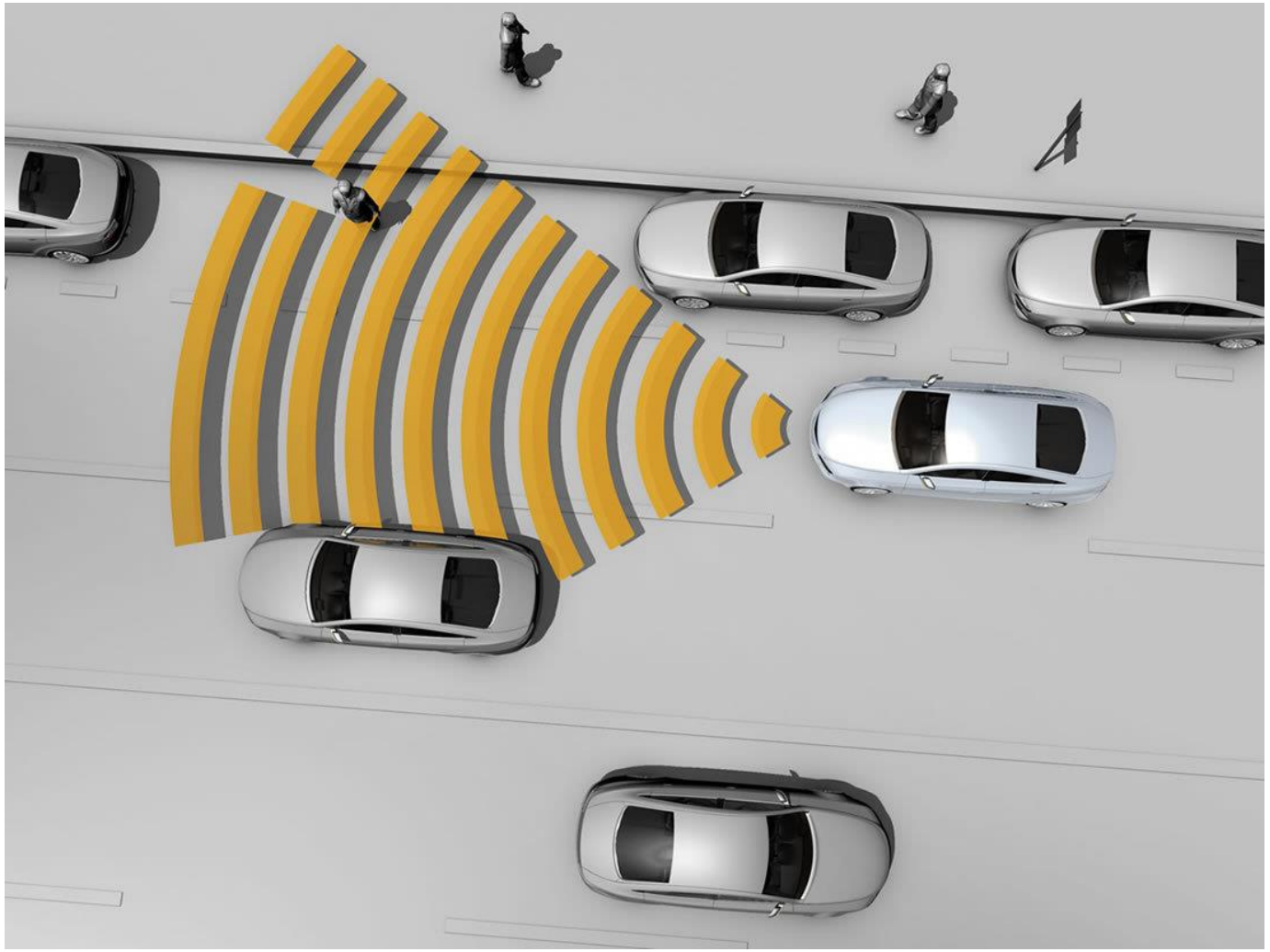


Geometria computacional: Encontrar obstáculos

computação
Universidade Federal de Pelotas

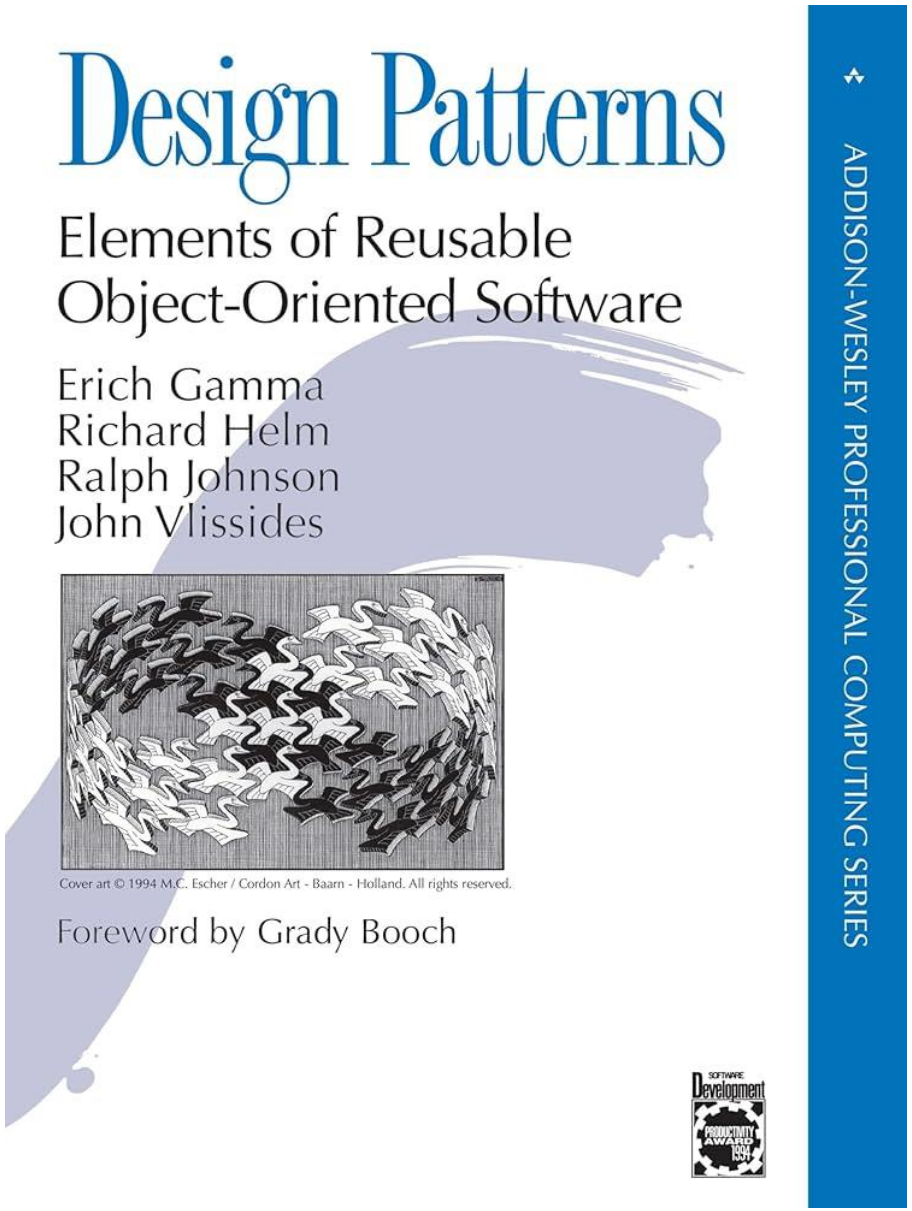


Envoltória convexa



Padrão de projeto

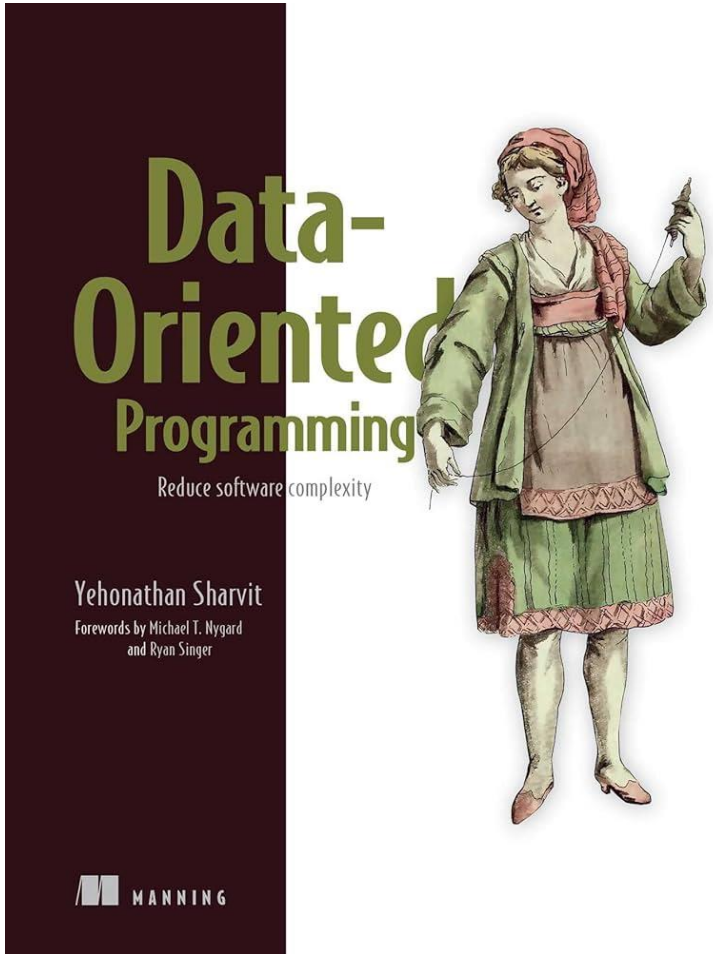
computação
Universidade Federal de Pelotas



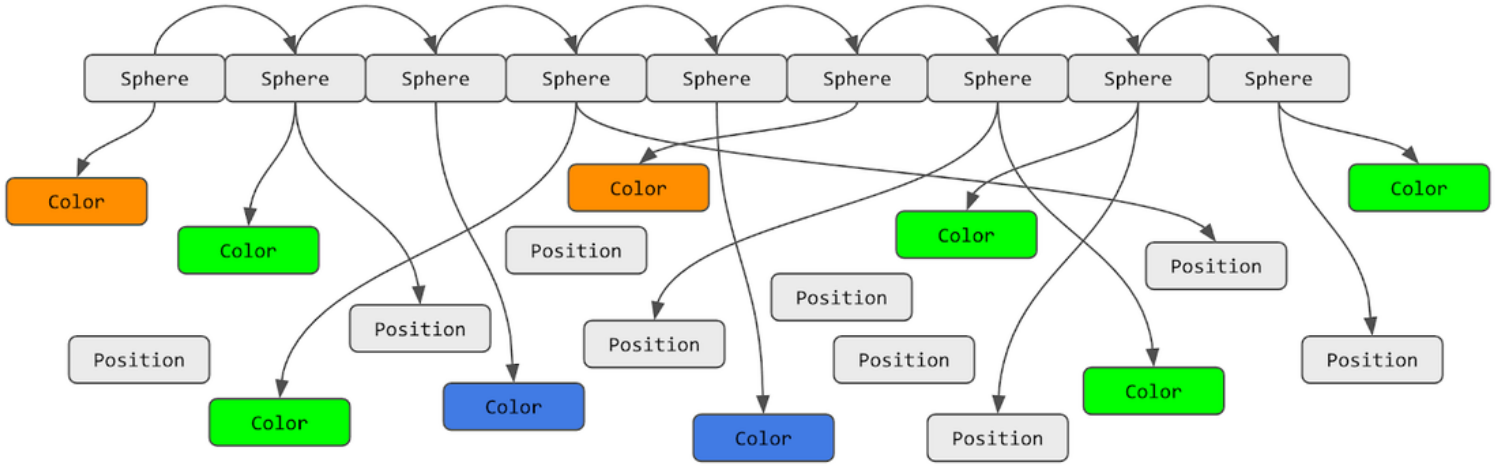
Creational Design Pattern	Structural Design Pattern	Behavioral Design Pattern
For handling Object creation mechanisms	For identifying ways to realize relationships between objects	For handling communication between different objects
Constructor	Adapter	Chain of Responsibility
Factory	Bridge	Command
Abstract Factory	Composite	Iterator
Prototype	Decorator	Mediator
Singleton	Facade	Memento
Builder	Flyweight	Observer
	Proxy	State
		Strategy
		Template method
		Visitor

Programação Orientada a Dados

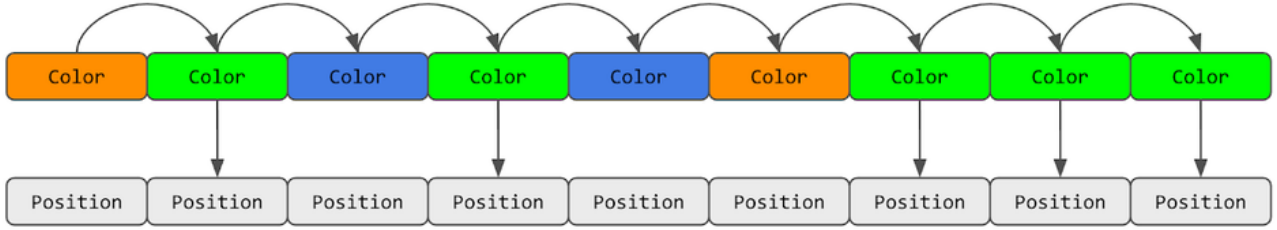
computação
Universidade Federal de Pelotas



OOP

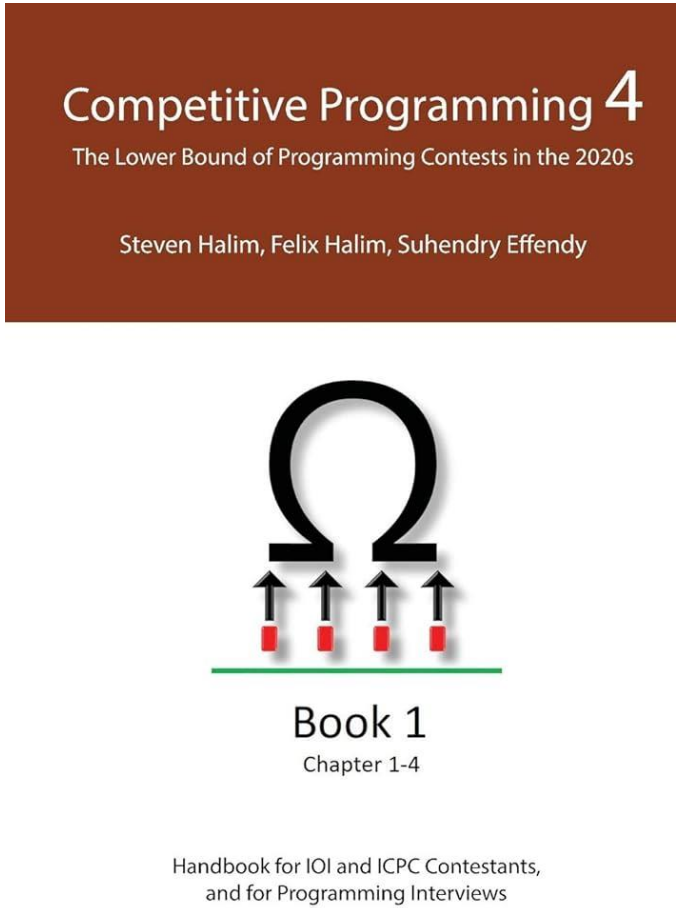


DOD



Técnicas de Programação

computação
Universidade Federal de Pelotas



- **Programação dinâmica** – resolução de subproblemas, técnicas de memoização e tabulação, com exemplos clássicos como o problema da mochila, subsequência crescente máxima e distância de edição.
- **Grafos** – métodos de representação (listas de adjacência, matrizes), algoritmos BFS, DFS, ordenação topológica, identificação de componentes fortemente conectados (Kosaraju, Tarjan).
- **Árvores e estruturas avançadas** – segment tree, fenwick tree (BIT), union-find (DSU), treap, splay tree, árvores AVL e Red-Black.
- **Fluxo máximo** – algoritmos de Edmonds-Karp e Dinic, min-cut e suas aplicações em emparelhamento bipartido.
- **Algoritmos de caminhos mínimos** – Dijkstra, Bellman-Ford, Floyd-Warshall, Johnson e A*.
- **Geometria computacional** – manipulação de pontos e vetores, produto vetorial, envoltória convexa (Graham scan, cadeia monótona de Andrew), interseção de segmentos, polígonos e cálculo de área.
- **Teoria dos números** – aritmética modular, exponenciação rápida, testes de primalidade, crivo de Eratóstenes, fatoração, funções de Euler e o teorema chinês do resto (CRT).
- **Combinatória** – técnicas de contagem, permutações e composições, princípio da inclusão-exclusão, geração de combinações e permutações, binômio de Newton.
- **Algoritmos para strings** – hashing (rolling hash), KMP, algoritmo Z, suffix array, suffix automaton, trie e Aho-Corasick.
- **Algoritmos avançados** – meet-in-the-middle, programação dinâmica com divisão e conquista, DP com bitmask, técnicas de binary lifting, algoritmo de Mo e consultas offline.
- **Heurísticas e meta-heurísticas** – backtracking com poda, branch-and-bound, beam search e introdução a algoritmos genéticos.
- **Problemas de otimização e matemática discreta** – programação linear (simplex básico), conceitos de teoria dos matroides, polinômios e transformada rápida de Fourier (FFT).
- **Técnicas práticas de implementação** – uso eficiente de bibliotecas padrão, gerenciamento rápido de entrada e saída, templates reutilizáveis e estratégias de depuração em competições.

Como serão as aulas

computação
Universidade Federal de Pelotas

- Apresentação de um conceito e um problema que precisa do conceito para ser resolvido
  - Inicio da aula: +- 1 hora
- Implementação da solução
  - A principio, pode escolher sua linguagem preferida, mas alguns problemas podem exigir linguagem orientada a objetos ou de baixo nível
  - Deve ser feito durante o restante da aula
  - Entrega por GIT

Formandos

computação
Universidade Federal de Pelotas

- Alerta aos formandos:
  - Por favor, não falar quem é formando!
  - Não dar dica que é formando, por exemplo, “estava fazendo o tcc e por isso...” ou “só falta essa disciplina” etc
  - Todos têm tratamento igual, inclusive os formandos

Avaliação

computação
Universidade Federal de Pelotas

- Vou somar todos os trabalhos solicitados em aula e calcular a média.
  - O objetivo é termos 1 trabalho por semana, então teremos algo em torno de 15 trabalhos
- Não tem prova!
- Entregas no seu git
  - A data indicada no git precisa respeitar o prazo dado para a tarefa!
- Você mesmo pode acompanhar a sua nota, pois a nota é 0 ou 10 para cada trabalho
  - Fez o trabalho **no prazo** é 10
  - **Para ser considerado entregue precisa ter feito tudo o que foi pedido**
  - Não tem nota para trabalho parcial, pois são muitos trabalhos e você pode recuperar.
  - **Durante a parte prática de cada aula vou solicitar de <u>alguns</u> que mostrem o que fez no trabalho anterior. Se não conseguir responder perguntas sobre o trabalho esse será zerado.**

Exame

computação
Universidade Federal de Pelotas

- O exame é composto por um trabalho prático abrangendo **todo o conteúdo da disciplina para ser feito em aula**

O objetivo do exame é avaliar todo o aprendizado do semestre, ou seja, equivale a todas as avaliações anteriores em conteúdo!
Por isso, sua densidade é alta!

IA e plágio

computação
Universidade Federal de Pelotas

- Pode usar IA e códigos de outros, você que decide se está aprendendo algo. Usar de forma correta não é problema! Porém, lembrem que irei perguntar sobre o trabalho na aula seguinte.
- Sugiro pensar que você vai competir no mercado contra quem sabe fazer e não depende de IA.
- Imagine ir na academia e todos os aparelhos de musculação são motorizados, você vai conseguir fazer tudo, mas vai sair com o que foi buscar?

Quem não sabe desenhar tem que pedir para a IA, deu nisso:

computação
Universidade Federal de Pelotas

Chamada

computação
Universidade Federal de Pelotas

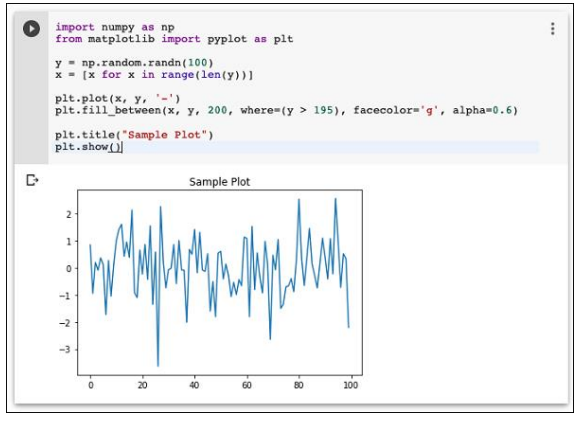
- No começo da aula.
- Se chegar depois, espere até o começo da parte prática da aula para falar comigo.
- Somente faço alterações em aula, se por algum motivo eu não puder fazer isso na aula, por exemplo, já fechei o notebook e estou saindo quando você lembra da chamada: me fale na aula e mande um email no mesmo dia para ficar registrado

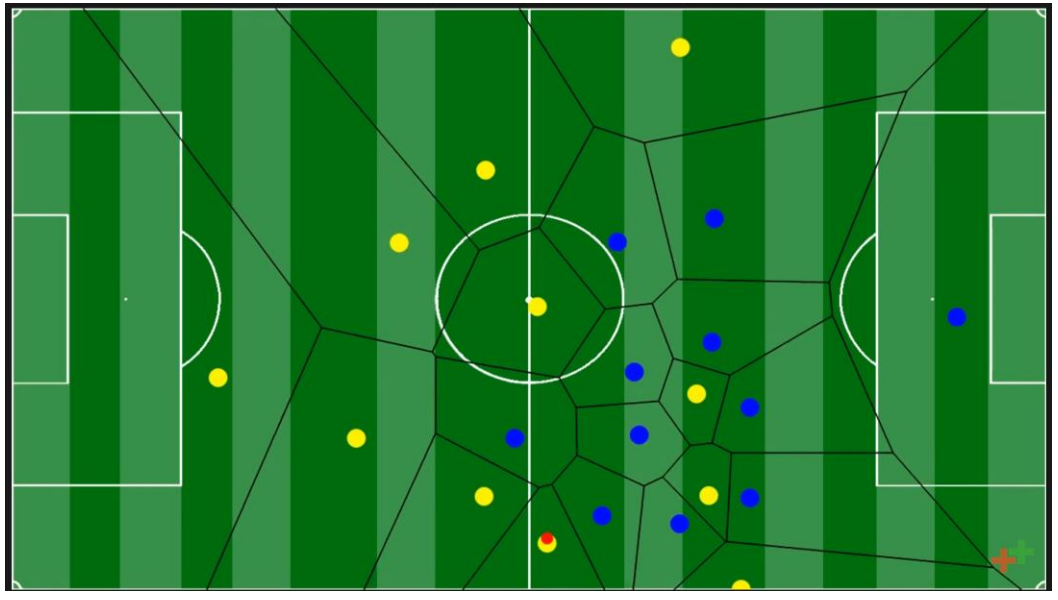
Trabalho 1: Gear Up!

computação
Universidade Federal de Pelotas

- **Montar um projeto que possa ser usado nos trabalhos seguintes**
  - Qualquer linguagem de programação
- O programa deve mostrar polígonos 2D e permitir que eles sejam alterados durante a execução.
  - Pontos, linhas, triângulos, polígonos
  - Definir cores por ponto, linha e área
  - Permitir clicar e encontrar a geometria onde foi clicado
- Gráficos de resultados
  - Colab ou qualquer outro que preferir
  - Configurar o seu projeto para exportar dados
  - Nesse caso, exporte um arquivo logando:
    - Todo o percurso do mouse na tela
    - Quantos cliques
    - Quais objetos clicou e em que momento
    - Tempo de execução
- **GIT**
  - Código
  - Vídeo dele rodando
  - Gráficos de uma execução
  - Código que gerou os gráficos
- **Prazo**: 24/9 23:59

https://www.youtube.com/watch?v=lA38E33un94





Trabalho 2: Diagrama de Voronoy

computação
Universidade Federal de Pelotas

- **Implementar diagrama de voronoy**
  - https://pt.wikipedia.org/wiki/Diagrama_de_Voronoy
  - Com o mouse clicar e criar pontos e o algoritmo calcula automaticamente o novo diagrama
  - Mostrar o gráfico dual do diagrama (Triangulação de Delaunay)
  - Descrava uma aplicação que acha interessante e que utiliza diagrama de voronoy, colocar isso num arquivo junto no git

- **GIT**
  - Código
  - Vídeo dele rodando
  - Faça alguns gráficos:
    - Custo computacional
    - Crescimento conforme mais pontos são adicionados
    - Existe diferença de custo computacional dependendo da distribuição de pontos?
    - Pense sobre o desempenho do algoritmo e tenta montar algum gráfico que demonstra o comportamento
  - Código que gerou os gráficos
- **Prazo**: 05/10 23:59

Criar a tabela de trabalhos e GIT

computação
Universidade Federal de Pelotas

- Colocar no readme do seu repositório da disciplina uma tabela listando os trabalhos feitos
- Para cada linha da tabela um link para a pasta do trabalho no seu GIT

Trabalho	Data e hora da entrega do último arquivo	Link para os arquivos no GIT	Fez tudo o que foi solicitado e no prazo?
1	17/9/25, 18:00		Sim
2			
…			
15			

Avisos

computação
Universidade Federal de Pelotas

- Venham conversar comigo se precisarem de ajuda!
- Façam isso logo, não esperem para o dia anterior a prova ou trabalho.
- Só posso ajudar se tiver tempo para isso.

- É comum falarem comigo só depois do final do semestre, mas aí já foi... Não adiante eu descobrir na avaliação da disciplina que só é liberada para o professor depois do final do semestre.

Como falar comigo:

1. rafael.torchelsen@inf.ufpel.edu.br
2. Ao final dos encontros
3. Discord da Computação nick Toto
4. Mensagem pelo e-aulas

Não enviar mensagem pelo Cobalto!